Міністерство освіти і науки України

Центральноукраїнський національний технічний університет

Механіко-технологічний факультет

ЗВІТ

ПРО ВИКОНАННЯ ЛАБОРАТОРНОЇ РОБОТИ № 11

з навчальної дисципліни

“Базові методології та технології програмування”

РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ

ЗАВДАННЯ ВИДАВ

доцент кафедри кібербезпеки та програмного забезпечення

Доренський О. П.

[https://github.com/odorenskyi/](https://github.com/odorenskyi/Dmytro-Parkhomenko-KB18)

ВИКОНАВ

студент академічної групи КІ-23

Чабан С.О.

ПЕРЕВІРИВ

ст. викладач кафедри кібербезпеки   
та програмного забезпечення

Дрєєва Г. М.

Кропивницький – 2024

**ТЕМА: РЕАЛІЗАЦІЯ ПРОГРАМНИХ ЗАСОБІВ ОБРОБЛЕННЯ ДИНАМІЧНИХ СТРУКТУР ДАНИХ ТА БІНАРНИХ ФАЙЛІВ**

**МЕТА:** Набути ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

**ЗАВДАННЯ:**

1. У складі команди ІТ-проекта розробити програмні модулі оброблення динамічної структури даних.

2. Реалізувати програмний засіб на основі розроблених командою ІТ-проекта модулів.

**СКЛАД КОМАНДИ:**

* Богданов Олександр
* Козирь Сергій
* Чабан Софія

**Варіант 9**

**Аналіз задач IT-проєкту та вимоги до ПЗ**

**Вимоги:**

1. Створення електронного довідника індексів та відділень поштового зв'язку України.
2. Забезпечення можливості виведення всього довідника на екран або у текстовий файл.
3. Автоматичне завантаження даних з файлу під час запуску програми.
4. Автоматичне збереження даних у файл під час завершення роботи програми.

**Аналіз та розподіл задач**

1. Пошук запису в довіднику за введеним індексом або назвою населеного пункта

Відповідальний за реалізацію - Козирь Сергій

1. Виведення всього довідника на екран або у текстовий файл

Відповідальний за реалізацію – Чабан Софія

1. Додавання нового запису в довідник

Відповідальний за реалізацію – Богданов Олександр

1. Вилучення заданого оператором запису з довідника

Відповідальний за реалізацію – Богданов Олександр

1. Завершення роботи програми з автоматичним записом даних у файл

Відповідальний за реалізацію – Чабан Софія

1. Дані довідника автоматично завантажуються з файлу під час запуску програми

Відповідальний за реалізацію - Козирь Сергій

**Архітектура:**

Файли:

struct\_type\_project\_9.h - заголовковий файл, що містить опис структури

prj\_4\_Сaban - результуюча програма.

ModulesChaban.a/.h (Модулі розроблені Чабан Софією)

ModulesBogdanov.a/.h (Модулі розроблені Богдановим Олександром)

ModulesKozir.a/.h (Модулі розроблені Козирем Сергієм)

**Лістинг заголовкового файлу struct\_type\_project\_2:**

#ifndef STRUCT\_TYPE\_PROJECT\_9\_H\_INCLUDED

#define STRUCT\_TYPE\_PROJECT\_9\_H\_INCLUDED

#include <string>

using namespace std;

struct Record {

string index;

string region;

string locality;

string street;

};

#endif // STRUCT\_TYPE\_PROJECT\_9\_H\_INCLUDED

**Лістинг модуля ModulesChaban:**

#include <fstream>

#include <sstream>

#include <iostream>

#include <vector>

#include <string>

#include <locale>

#include <codecvt>

#include "ModulesChaban.h"

using namespace std;

// Функція для відображення реєстру

void displayRegistry(const vector<Record>& registry) {

for (const auto& record : registry) {

cout << record.index << " "

<< record.region << " "

<< record.locality << " "

<< record.street << endl;

}

}

// Функція для збереження даних у файл

void saveData(const vector<Record>& data, const string& filename) {

ofstream outFile(filename);

if (outFile.is\_open()) {

for (const auto& record : data) {

outFile << record.index << " "

<< record.region << " "

<< record.locality << " "

<< record.street << endl;

}

outFile.close();

cout << "Реєстр збережено у файл." << endl;

} else {

cout << "Неможливо відкрити файл для запису: " << filename << endl;

}

}

**Лістинг заголовкового файлу ModulesChaban:**

#ifndef MODULESCHABAN\_H\_INCLUDED

#define MODULESCHABAN\_H\_INCLUDED

#include <iostream>

#include <vector>

#include <string>

#include <fstream>

void saveData(const vector<Record>& data, const string& filename);

void displayRegistry(const vector<Record>& registry);

#endif // MODULESCHABAN\_H\_INCLUDED

**Лістинг prj\_9\_Chaban:**

#include <iostream>

#include <fstream>

#include <vector>

#include <string>

#include "struct\_type\_project\_9.h"

#include "MoudelsChaban.h"

#include "MoudelsKozyr.h"

#include "MoudelsBogdanov.h"

using namespace std;

int main() {

string filename = "index.txt";

vector<Record> registry = loadData(filename);

// Основний цикл програми

int option;

do {

cout << "Меню:\n";

cout << "1. Додати запис\n";

cout << "2. Видалити запис\n";

cout << "3. Шукати запис за поштовим кодом\n";

cout << "4. Завантажити реєстр з файлу\n";

cout << "5. Вивести реєстр\n";

cout << "6. Зберегти реєстр\n";

cout << "7. Вийти з програми\n";

cout << "Виберіть опцію: ";

cin >> option;

switch (option) {

case 1:

addRecord(registry);

break;

case 2:

deleteRecord(registry);

break;

case 3:

searchRecord(registry);

break;

case 4:

registry = loadData(filename);

cout << "Реєстр перезавантажено з файлу." << endl;

break;

case 5:

displayRegistry(registry);

break;

case 6:

saveData(registry, filename);

break;

case 7:

cout << "Вихід з програми." << endl;

break;

default:

cout << "Неправильна опція. Спробуйте ще раз." << endl;

break;

}

} while (option != 7);

return 0;

}

Висновки:

У ході виконання лабораторної роботи № 11, опрацювала роботу с бінарними файлами та динамічними структурами даних. Набула ґрунтовних вмінь і практичних навичок командної (колективної) реалізації програмного забезпечення, розроблення функцій оброблення динамічних структур даних, використання стандартних засобів С++ для керування динамічною пам’яттю та бінарними файловими потоками.

В файл README, записати тему, мету, варіант та завдання. Зібралися командою та обговорили:

Аналіз задачі та вимог до неї.

Визначили вид динамічної структури для реалізації бази даних ПЗ, ми обрали тип vector.

Створили заголовковий файл struct\_type\_project\_9.h, в ньому описали структуру бази даних.

Розділили задачі між учасниками команди.

Склали план роботи з виконання ІТ-проєкта, виходячи із розподілених між учасниками команди субзадач.

Завантажила всі файли на Git-репозиторій.